Smoothed Analysis of Algorithms

by Eklavya Sharma

Part 1:
Intro to Smoothed Analysis

Analysis of Algorithms

- Analyzing an algorithm is an attempt to predict its performance.

- 'Performance' can mean running time, quality of solution, etc.

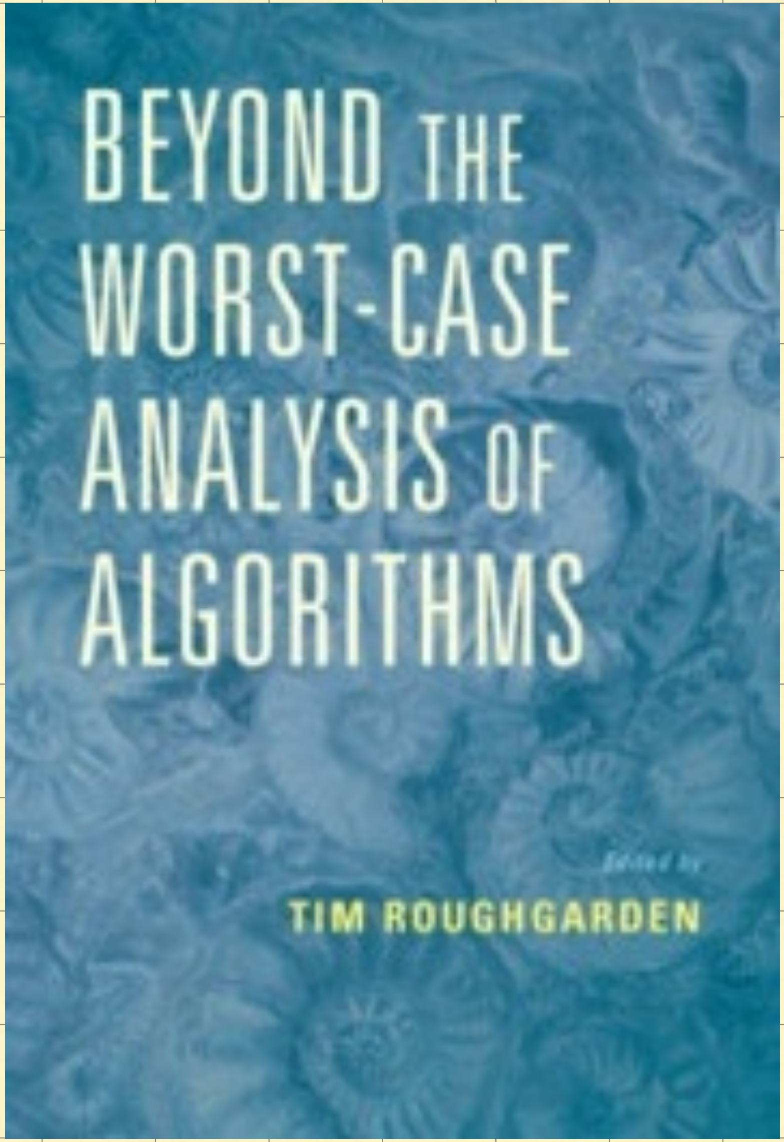
- We often use worst-case analysis.
  - Great for positive results.
  - Not always good for negative results.

Problems with worst-case analysis

- E.g., simplex method for LP:
  - Works well in practice.
  - Known hard input: Klee-Minty cube: n vars, n constraints, (most variants of) simplex method take $\Omega(2^n)$ time.
- Worst-case input often doesn't occur in practice.
- Negative worst-case results can be misleading.
- Many such examples are known.

So what can we do?

1. Parametric Analysis
2. Resource Augmentation
3. Semi-random models
4. Smoothed Analysis
5. Other techniques?



Adversary in worst-case analysis

Worst-case analysis is equivalent to assuming the presence of an adversary.

1. We pick algorithm A for a problem.
2. Adversary picks input $i \in I$. (i can depend on A) such that $c_A(i)$ is maximized.

Is it reasonable to assume we have an adversary?

Sometimes, yes. Please use worst-case analysis, else you're susceptible to ACA (algorithmic complexity attack).

But often there's no adversary.

Assumption: Input comes from a distribution.
We don't know the distribution, but we assume something reasonable about the distribution (e.g. lower bound on variance).

Smoothed Analysis

Intuition: Adversary has a 'trembling hand'.
In many applications, input is inherently noisy.

I: set of inputs. For $i \in I$, $c(i)$ is cost on i.

Let $s: I \to I$ be a randomized function, called the smoothing function.
For each $i \in I$, $s(i)$ defines a distribution over inputs.

The s-smoothed cost is

$$\max_{\underbrace{i \in I}_{\text{what adversary wanted to pick}}} \; \mathbb{E}_{\underbrace{\hat{i} \sim s(i)}_{\text{what adversary ended up picking}}} c(\hat{i}).$$

$\hat{i}$ is called the perturbed input.

Examples of smoothing functions

1. $\max_x c^T x$ where $Ax \le b$. (where $\|(a_i, b_i)\|_2 \le 1 \ \forall i$)
   add IID $N(0,\sigma)$ noise to each entry of A and b.

2. Bin packing: n boxes of weights $w_1, w_2, \ldots, w_n$.
   Pack into min no. of shipping containers of capacity 1.
   $s(w) := \hat{w}$, where $\hat{w}_i := g(w_i z_i)$.
   $g(x) = \begin{cases} x & \text{if } x \in [0,1] \\ 0 & \text{if } x \le 0 \\ 1 & \text{if } x \ge 1 \end{cases}$. $z_1, z_2, \ldots, z_n$ are IID $N(1, \sigma)$.

When should we try smoothed analysis?

1. Worst-case results are much worse than empirical observations.
2. Input sources are noisy/random (e.g. computer vision)

Especially good to try if

1. We know good algorithms for special cases.
2. Hard inputs are brittle.
3. Average-case analysis gives good results.

Applications of Smoothed Analysis

1. Simplex Algorithm (Spielman, Teng, STOC 01)
2. 2-opt heuristic for TSP (ERV, SODA 07)
3. GBP and VBP with additive noise (Karger, Onak, SODA 07)
4. K-means (AMR, FOCS 09)
5. Perceptron (Avrim Blum, John Dunagan, SODA 02) (Today's talk)

Template for applying smoothed analysis

1. Show that the algorithm does well for special cases.
2. Inputs outside special cases are called ‘bad’. Study their properties.

   Prove that they’re ‘brittle’, i.e., changing input slightly will violate them.
3. Show that smoothed inputs are unlikely to satisfy those properties.

Useful tool from probability : Results like $P(X \in S) \leq \varepsilon$ for some set S and randvar X from a relevant prob distr.
(E.g. tail bounds, anti-conc bounds)

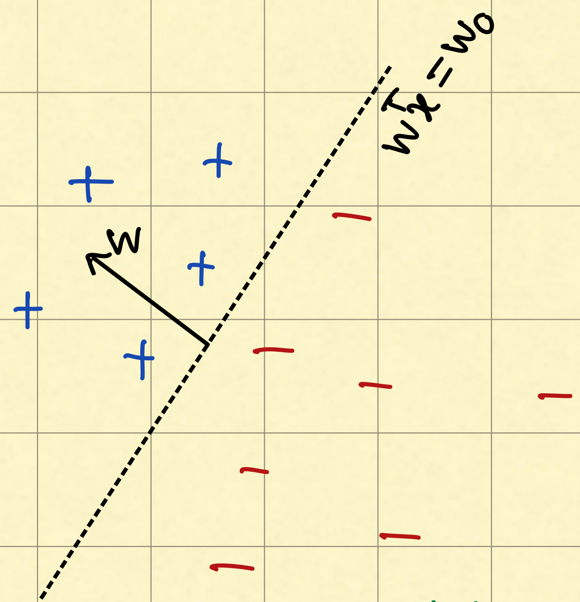
Different template for smoothed analysis

1. Show that smoothed instances satisfy certain properties (whp).
2. Show that the algorithm does well when input has those properties.

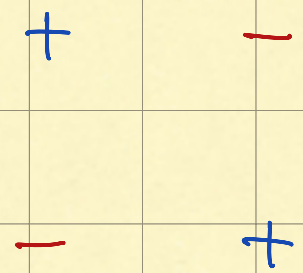
Part 2:
Perceptron Algorithm

Problem A: (Linear classification)

Given vectors $a_1, a_2, \ldots, a_n \in \mathbb{R}^d$ and labels $y_1, y_2, \ldots, y_n \in \{-1, 1\}$, find (if exists) $w \in \mathbb{R}^d$, $w_0 \in \mathbb{R}$ s.t. $\forall i$, $\left(\begin{array}{l} a_i^T w > w_0 \text{ if } y_i = 1 \\ \& \ a_i^T w < w_0 \text{ if } y_i = -1 \end{array} \right)$.



solution exists

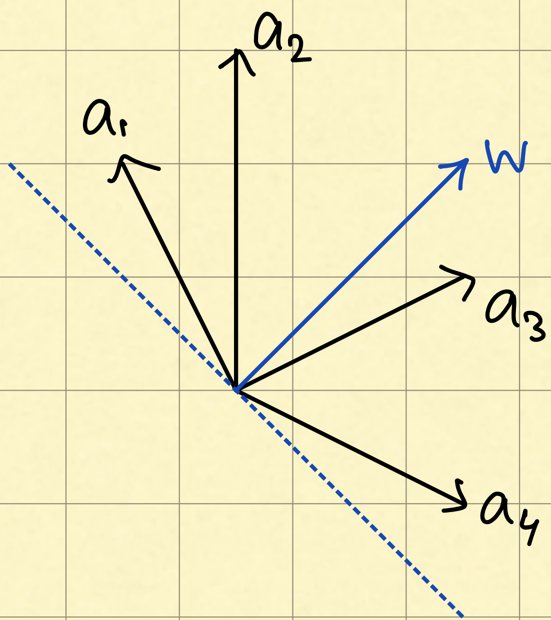


no solution

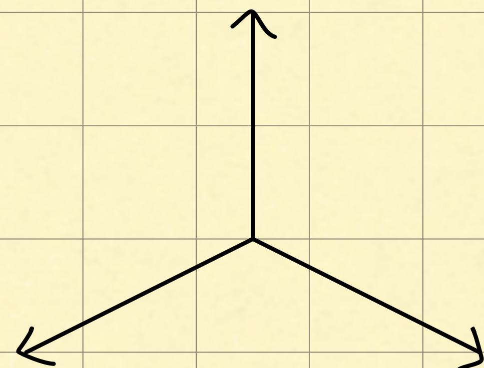
(then algorithm allowed to do anything)

Problem B: (Finding a popular direction)

Given $a_1, \ldots, a_n \in \mathbb{R}^d$, find $w \in \mathbb{R}^d$ s.t. $a_i^T w > 0 \ \forall i \in [n]$.



solution exists



no solution exists

(then algo allowed to do anything)

Reducing LinClass to PopDir

(w, w_0) is soln to LinClass inst $[(a_i, y_i)]_{i=1}^n$ $\iff$ $\left\{\begin{array}{ll} a_i^T w > w_0 & \text{if } y_i = 1 \\ a_i^T w < w_0 & \text{if } y_i = -1 \end{array}\right\} \forall i \in [n]$

$$\Updownarrow$$

$$\underbrace{[y_i a_i^T, -y_i]}_{b_i} \underbrace{\begin{bmatrix} w \\ w_0 \end{bmatrix}}_{v} > 0 \quad \forall i \iff (a_i^T w - w_0) y_i > 0 \quad \forall i \in [n]$$

$$\Updownarrow$$

v is a solution to PopDir instance $[b_i]_{i=1}^n$.

$\therefore$ We will focus on PopDir. Forget about LinClass.

Polytime algorithm for PopDir?

(Recall: Given $a_1, \dots, a_n \in \mathbb{R}^d$ find $w \in \mathbb{R}^d$ s.t. $a_i^T w > 0 \ \forall i$.)

$$\left(\exists\, w \in \mathbb{R}^d \text{ s.t. } a_i^T w > 0 \ \forall i\right) \iff \left(\exists\, \hat{w} \in \mathbb{R}^d \text{ s.t. } a_i^T \hat{w} \geq 1 \ \forall i\right)$$

Perceptron Algorithm

Works well in practice for PopDir.

1. Set $w = 0$.
2. while $\exists\, i \in [n]$ s.t. $a_i^T w \leq 0$:
3. $\quad w = w + \dfrac{a_i}{\|a_i\|}$
4. return w

Does the perceptron algorithm terminate? How quickly?

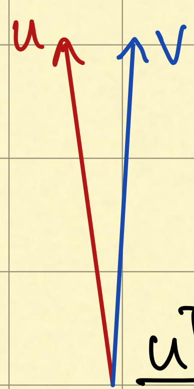
Known results:

1. (Block, Novikoff, 1962) Always terminates.
2. There is an example where it takes exponential time.
   (Papers say this is 'known' and 'easy to see' but I couldn't figure it out or find a reference.)

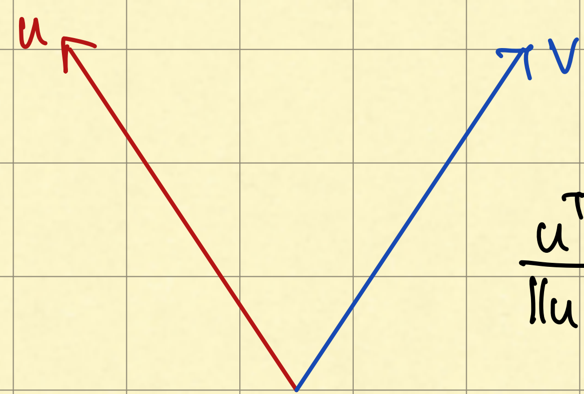
In practice, it usually terminates quickly.
Can we explain this using smoothed analysis?

Interpreting the dot product.

Let $u, v \in \mathbb{R}^d$. $\dfrac{u^T v}{\|u\| \|v\|} \in [-1, 1]$ tells us how similar their directions are

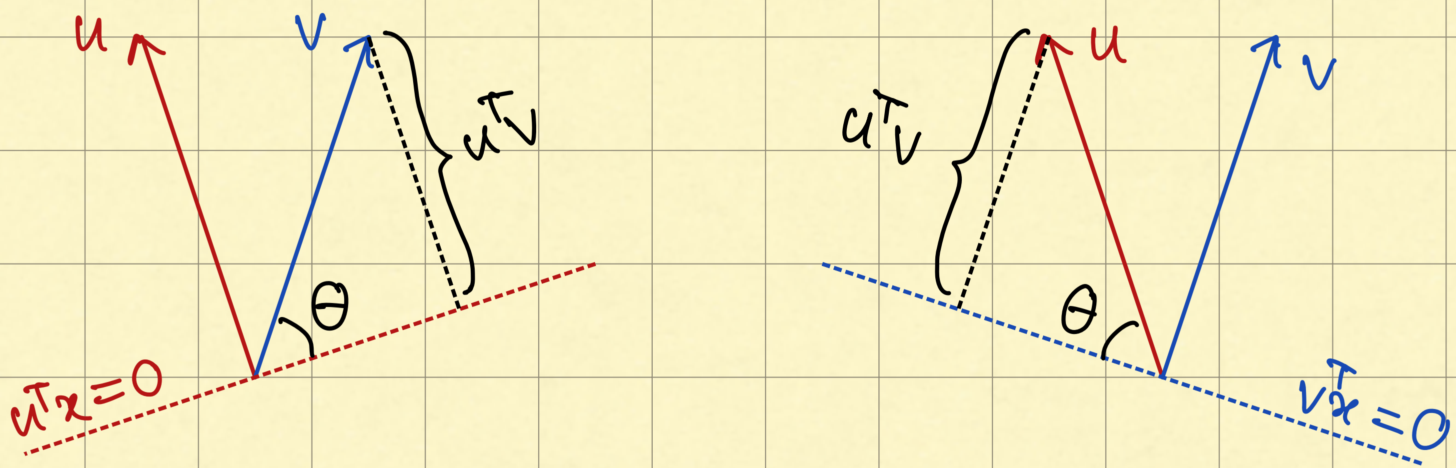


$\dfrac{u^T v}{\|u\| \|v\|}$ is large



$\dfrac{u^T v}{\|u\| \|v\|}$ is small

Let u, v be unit vectors.

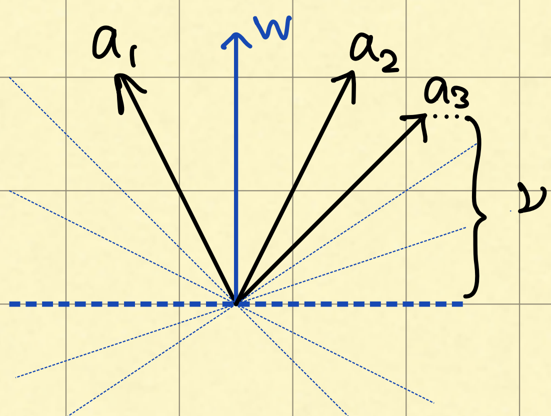


$$u^T v = \sin\theta$$

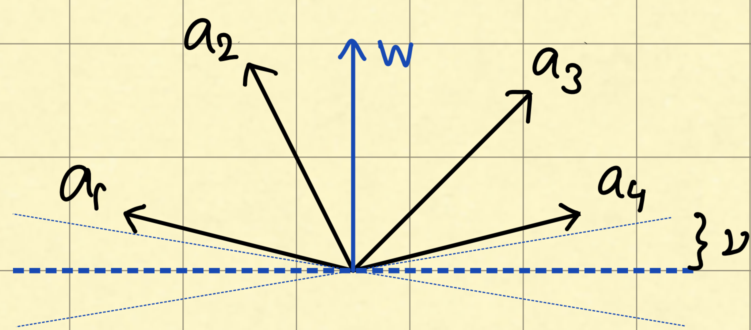
Step 1 : Special case analysis

Wiggle room : For inputs $a_1, \dots, a_n$ and soln w,

$$\nu(w) := \min_{i=1}^{n} \frac{a_i^T w}{\|a_i\| \|w\|} .$$



(large wiggle)



(small wiggle)

Perceptron Convergence Theorem (Block, Novikoff, 1962)

Suppose $\exists\, w^* \in \mathbb{R}^d$ of wiggle room $\nu > 0$. Then the perceptron algorithm terminates in $\leq \lfloor 1/\nu^2 \rfloor$ iterations.

Proof. (main idea: $w^T w^* / \|w\| \|w^*\|$ increases)

Suppose we change w to $w + a_i$. (WLOG $\|a_i\| = \|w^*\| = 1$)

Then (i) $(w + a_i)^T w^* = w^T w^* + a_i^T w^* \geq w^T w^* + \nu$.

(ii) $\|w + a_i\|^2 = \|w\|^2 + 1 + 2 a_i^T w \leq \|w\|^2 + 1$.

$\therefore$ After T iterations, $w^T w^* \geq T\nu$ and $\|w\| \leq \sqrt{T}$.

$T\nu \leq w^T w^* \leq \|w\| \leq T \implies T \leq 1/\nu^2$. $\square$

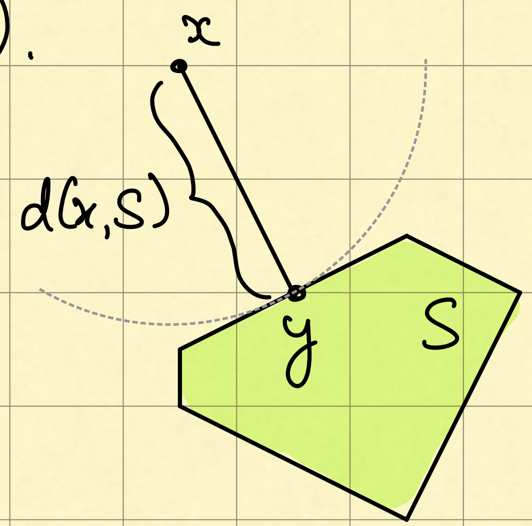
Part 3:
Smoothed Analysis

Smoothed Analysis.

1. (Done) Special case: good algorithm for high-wiggle.
2. Study properties of low-wiggle instances.
3. Show that smoothed instances will probably not satisfy those properties, which means wiggle is high.

Distances

$$d(x,y) = \|x-y\|, \quad d(x,S) = \inf_{y \in S} d(x,y).$$



Lemma: $d(z, \{x : a^T x = 0\}) = \dfrac{a^T z}{\|a\|}$.

Proof sketch. Defn of d with method of Lagrange's multipliers.

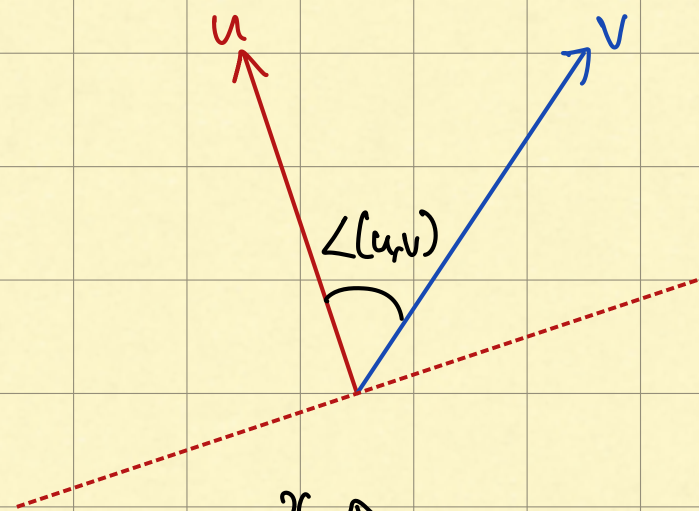
Angles

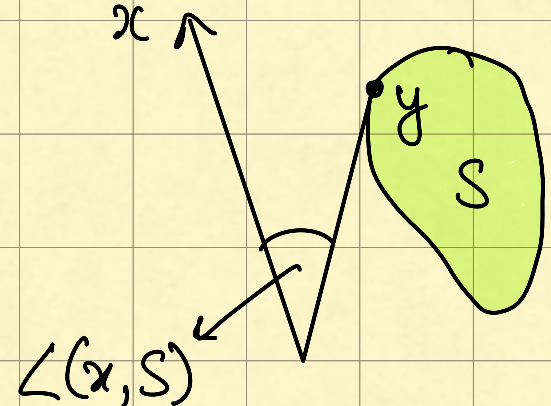
$$\angle(u,v) = \frac{\pi}{2} - \sin^{-1}\left(\frac{u^T v}{\|u\|\,\|v\|}\right)$$

$$\angle(x,S) = \inf_{y \in S} \angle(x,y).$$

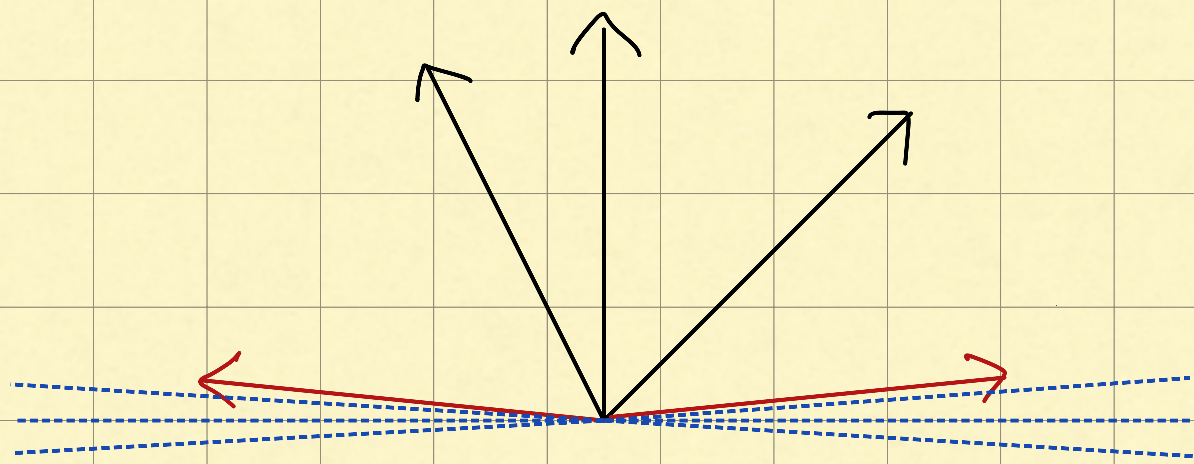
<u>Lemma</u>: $\angle(w, \{x : a^T x = 0\})$

$$= \sin^{-1}\left(\frac{a^T w}{\|a\|\,\|w\|}\right)$$





Low wiggle implies bad vectors



Bad vectors are also brittle

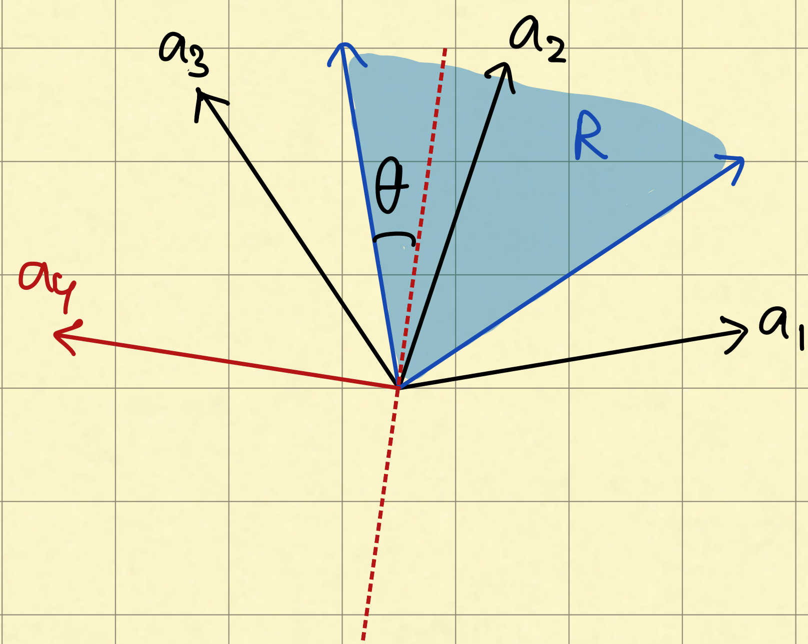
(perturbing them will make them good, or problem infeas)

Goodness

Fix $i \in [n]$. We want to define whether a_i is good/bad.
Goodness is relative to $\{a_j : j \neq i\}$. So fix $a_j \ \forall j \neq i$.

$R := \{w : a_j^T w > 0 \ \forall j \neq i\}$. (semi-feasible solutions)
If $R = \phi$, then input is infeas regardless of a_i.
So assume $R \neq \phi$.
$W := \{w : a_j^T w > 0 \ \forall j\}$ (feasible solutions) $(W \subseteq R)$
$H_j := \{x : a_j^T x = 0\}$ (hyperplane perpendicular to a_j)

Example with $i=4$.



R is almost polyhedral:

$a_j^T w > 0$ defines open halfspace.

Possibilities?

1. a_i makes problem infeas.
2. a_i keeps problem feas.

$$\text{good}(a_i) := \sup_{w \in R} \frac{a_i^T w}{\|a_i\|\|w\|} \quad \left(= \sup_{w \in W} \sin \angle (H_i, w) \text{ when } W \neq \emptyset \right)$$

Pick $\varepsilon > 0$. $\text{good}(a_i)$: $\begin{cases} \leq 0 \iff \text{infeas} \\ (0, \varepsilon] \implies \varepsilon\text{-bad} \\ > \varepsilon \implies \varepsilon\text{-good} \end{cases}$

Overview

1. Show that low wiggle $\Longrightarrow$ some vectors are bad.
2. After smoothing, $\forall i$, a_i is unlikely to be bad.
3. By union bound, no vector is bad w.h.p.

$\Longrightarrow$ high wiggle $\Longrightarrow$ quick termination.

<u>Theorem</u>: Let ν be the max wiggle room, i.e.,

$$\nu := \max_{w \in \mathbb{R}^d - \{0\}} \min_{i \in [n]} \frac{a_i^T w}{\|a_i\| \|w\|}.$$

Then $\min_{i \in [n]} \text{good}(a_i) \leq \frac{(d+1)\nu}{\sqrt{1-\nu^2}}$ (so small wiggle $\Longrightarrow$ some vector is bad)

$$\text{good}(a_i) := \sup_{w \in \mathbb{R}} \frac{a_i^T w}{\|a_i\| \|w\|}.$$

(Note: argsup can be different for each a_i)

$\min_{i=1}^{n} \text{good}(a_i)$ is large $\Longleftrightarrow$ $\forall i, \exists w_i \in W, \frac{a_i^T w_i}{\|a_i\| \|w_i\|}$ is large

$\forall\exists$ changes to $\exists\forall$

ν is large $\Longleftrightarrow$ $\exists w^*, \forall i, \frac{a_i^T w^*}{\|a_i\| \|w^*\|}$ is large

Theorem: Let ν be the max wiggle room, i.e.,

$$\nu := \max_{w \in \mathbb{R}^d - \{0\}} \min_{i \in [n]} \frac{a_i^T w}{\|a_i\| \|w\|}.$$

Then $\displaystyle \min_{i \in [n]} \text{good}(a_i) \le \frac{(d+1)\nu}{\sqrt{1-\nu^2}}$ (so small wiggle $\Longrightarrow$ some vector is bad)

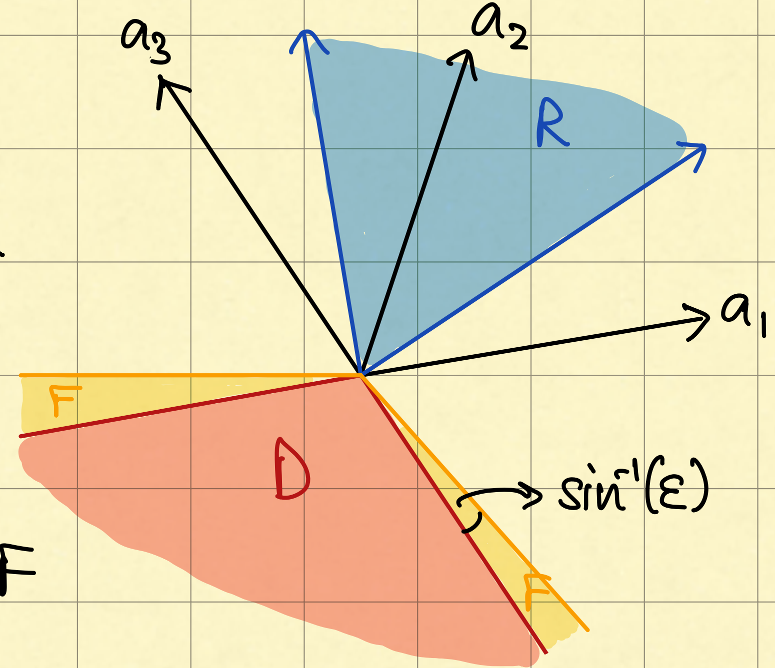
- Proof is too big too cover here.
- Proof has nice ideas about cones and convexity.
- Proof is problem-specific; not quite illustrative of how smoothed analysis works.
- Proof seems to have an error.

Let $R := \{w : a_j^T w > 0 \ \forall j \neq i\}$. (semi-feasible solutions)

$$\mathrm{good}(a_i) := \sup_{w \in R} \frac{a_i^T w}{\|a_i\| \|w\|} \quad \left(= \sup_{w \in W} \sin \angle(H_i, w) \text{ when } W \neq \emptyset \right)$$

Let $D := \{x : x^T w \leq 0 \ \forall w \in R\}$ (invalid input vectors)

$$= \{a_i : \mathrm{good}(a_i) \leq 0\}$$



Let $F := \{x : \angle(x, D) \leq \sin^{-1}(\varepsilon)\} - D$.

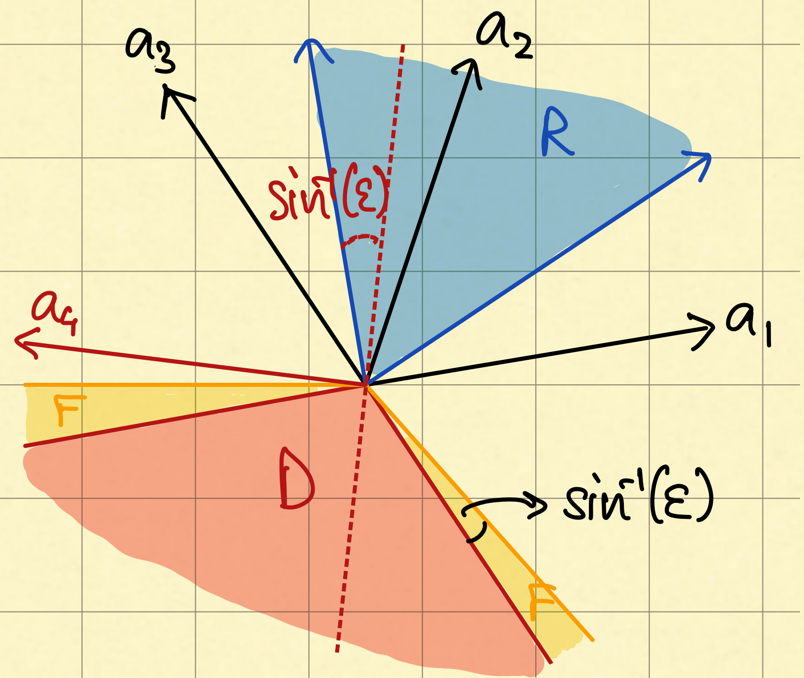
(ε-angular envelope of D)

<u>Lemma</u>: $\mathrm{good}(a_i) \in (0, \varepsilon] \iff a_i \in F$

Proof. 2 directions

1. $\text{good}(a_i) \in (0, \varepsilon] \implies a_i \in F.$

2. $\text{good}(a_i) > \varepsilon \implies a_i \notin F.$



(1.) $\text{good}(a_i) > 0 \implies a_i \notin D$

$$a_i \xrightarrow{\text{rot by } \sin^{-1}(\varepsilon)} \hat{a}_i$$

$$\text{good}(a_i) = \sup_{w \in W} \sin \angle(H_i, w)$$

$\angle(a_i, \hat{a}_i) \leq \sin^{-1}(\varepsilon).$

$\hat{a}_i \in D \implies \angle(a_i, D) \leq \sin^{-1}(\varepsilon)$

$\implies a_i \in F.$

(2. $\mathrm{good}(a_i) > \varepsilon \implies a_i \notin F$.) $\angle(u,v) := \frac{\pi}{2} - \sin^{-1}\left(\frac{u^T v}{\|u\|\|v\|}\right)$.

$$\mathrm{good}(a_i) = \sup_{w \in R} \frac{a_i^T w}{\|a_i\|\|w\|} > \varepsilon \implies \exists\, \hat{w} \in R,\ \frac{a_i^T \hat{w}}{\|a_i\|\|\hat{w}\|} > \varepsilon.$$

$$\implies \angle(a_i, \hat{w}) < \frac{\pi}{2} - \sin^{-1}(\varepsilon).$$

Pick any $x \in D$. Then $x^T \hat{w} \le 0$. So $\angle(x, \hat{w}) \ge \pi/2$.

$$\angle(x, a_i) \ge \angle(x, \hat{w}) - \angle(\hat{w}, a_i) > \pi/2 - (\pi/2 - \sin^{-1}(\varepsilon)) = \sin^{-1}(\varepsilon)$$

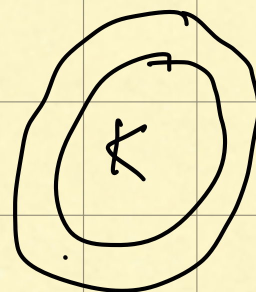
$$\implies \angle(a_i, D) > \sin^{-1}(\varepsilon) \implies a_i \notin F. \qquad \square$$

Smoothing

$s([a_1, \dots, a_n]) = [a_1', a_2', \dots, a_n']$, where

$a_i' := a_i + \|a_i\| z_i$ and $z_1, \dots, z_n$ are IID $N(0,\sigma)^d$

WLOG $\|a_i\| = 1$ $\forall i \in [n]$. ($\not\Rightarrow \|a_i'\| = 1$)



<u>Lemma</u> (small boundaries are easily missed)

Let $K \subseteq \mathbb{R}^d$ be a convex set.

Let $\Delta(K,\varepsilon) := \{x : d(x,K) \leq \varepsilon\} - K$. ($\varepsilon$-boundary of K)

Let $z \sim N(\mu,\sigma)^d$. Then $P(z \in \Delta(K,\varepsilon)) \in O\left(\frac{\varepsilon\sqrt{d}}{\sigma}\right)$.

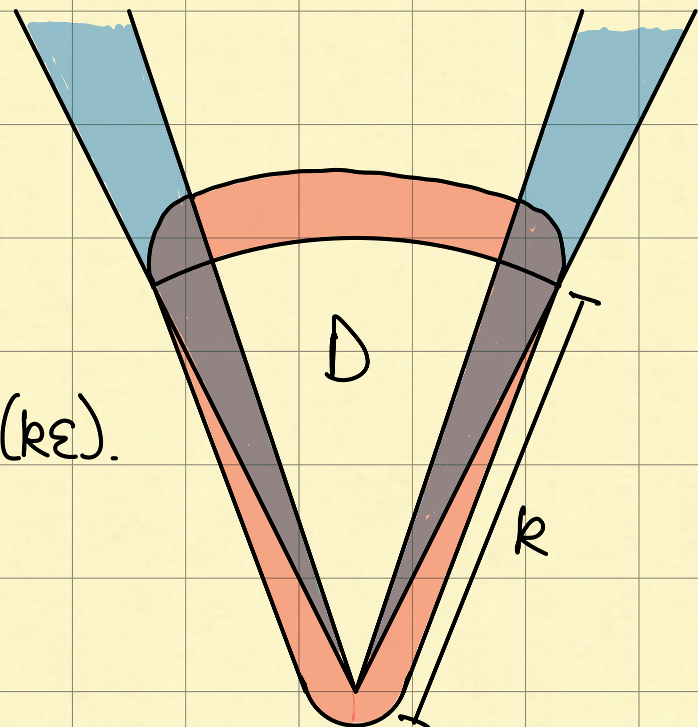
We showed $\text{good}(a_i') \in (0, \varepsilon] \iff a_i' \in F$.

$F = \{x : \angle(x, D) \leq \sin^{-1}(\varepsilon)\}$.

For some k, $\|a_i'\| \leq k$ w.h.p. (tail bd)

Let $B = \{x : \|x\| \leq k\}$.

Then $F \cap B \subseteq \Delta(D \cap B, \tau)$ for $\tau \in O(k\varepsilon)$.



$$P(\text{good}(a_i') \in (0,\varepsilon]) = P(a_i' \in F)$$

$$= P(a_i' \in F \cap \overline{B}) + P(a_i' \in F \cap B)$$

$$\leq \underbrace{P(a_i' \notin B)}_{\text{low}} + \underbrace{P(a_i' \in \Delta(D \cap B, \tau))}_{\text{low}}$$

F vs $\Delta(D \cap B, \tau)$

Precise results.

$\forall i,\ P(\mathrm{good}(a_i') \in (0,\varepsilon]) \in O(\alpha^{1/4} \log(1/\alpha))$ $\quad \alpha := \varepsilon\sqrt{d}/\sigma$

$(\&\ \sigma^2 \leq 1/2d)$

$\mathrm{good}(a_i') > \varepsilon\ \forall i \implies \nu \geq \frac{\varepsilon}{2(d+1)}$.

$\implies \#\text{iterations} \in O(d^2/\varepsilon^2)$.

Let $\delta > 0$. Set $\varepsilon = O\left(\frac{\sigma}{\sqrt{d}}\left(\frac{\delta}{n}\right)^4 \frac{1}{\log^4(n/\delta)}\right)$ to get that

with prob $1-\delta$, $\#\text{iters} \in O\left(\frac{d^3}{\sigma^2}\left(\frac{n}{\delta}\log\left(\frac{n}{\delta}\right)\right)^8\right)$ (if input is feasible)

Thank you.
Questions?